JURNAL MATEMATIKA DAN KOMPUTER
Vol. 5. No. 3, 168 - 177, Desember 2002, ISSN : 1410-8518

SOLUSI MASALAH KNAPSACK COLLAPSING DENGAN PENDEKATAN MASALAH KNAPSACK BAKU DAN PROGRAM DINAMIK [1]

Indarsih dan Widodo

Jurusan Matematika FMIPA Universitas Gadjah Mada

Abstrak

Masalah knapsack collapsing merupakan generalisasi masalah knapsack 0-1. Kapasitas pada masalah knapsack collapsing merupakan fungsi turun (non-increasing function) atas jumlah item yang dimuat. Masalah ini dapat diselesaikan dengan dua pendekatan, yaitu : reduksi ke masalah knapsack 0-1 dan pendekatan program dinamik. Masalah knapsack collapsing dapat diselesaikan dalam waktu pseudo-polinomial dengan kedua algoritma tersebut. Kedua algoritma tersebut akan diimplementasikan ke program dalam bahasa Pascal/C. Data instance yang diberikan akan dijalankan pada kedua program tersebut. Kemudian waktu tempuhnya dicatat. Eksperimen komputasi membuktikan bahwa kedua algoritma tersebut efisien.

Kata kunci : masalah knapsack collapsing, masalah knapsack 0-1, program dinamik.

1. PENDAHULUAN

Masalah knapsack collapsing (MKC) termasuk masalah knapsack non linear. Kapasitasnya merupakan fungsi tidak bertambah (*non-increasing function*) atas jumlah item yang termuat. MKC pertama kali dikemukakan oleh Posner (1978). Penelitian ini dimotivasi oleh aplikasi dalam satelit komunikasi. Pada saat hanya ada satu pengguna seluruh jalur komunikasi dapat digunakan. Akan tetapi jika ada pengiriman ganda pada jalur komunikasi maka diperlukan partisi penugasan jalur untuk masing-masing pengguna. Partisi ini mencegah pembicaraan silang tetapi juga mengefektifkan kapasitas komunikasi. Diberikan

[1] Disampaikan dalam Seminar Nasional dan Konferda Matematika ke-8 di UNDIP Semarang, 9 Maret 2002.

kendala kapasitas pada masalah sehingga para pengguna diijinkan untuk berkomunikasi. Fungsi yang dimaksimumkan merupakan jumlahan nilai para pengguna. Bobot merupakan ukuran jalur yang dibutuhkan oleh pengguna dan kapasitas jalur merupakan fungsi dari jumlah pengguna.

Pferschy dkk (1995) menyelesaikan MKC dengan mereduksinya ke masalah knapsack baku (RMKB) dan melalui pendekatan pemrograman dinamik (PPD). Dalam RMKB setelah MKC direduksi ke masalah knapsack 0-1 (MKB), masalah diselesaikan dengan algoritma minknap yang dikemukakan oleh Pisinger (1995). MKC dapat diselesaikan dalam waktu pseudo-polinomial melalui program dinamik. Ditinjau dari kompleksitasnya waktu tempuh algoritma PPD lebih kecil daripada waktu tempuh algoritma RMKB.

Penelitian ini bertujuan untuk membuktikan bahwa algoritma RMKB dan PPD efisien dan mengetahui waktu tempuh kedua algoritma tersebut dalam menyelesaikan suatu masalah jika dioperasikan dengan komputer. Penelitian ini dilakukan dengan cara mengimplementasikan algoritma RMKB ke program dalam bahasa C versi 2.0 dan algoritma PPD akan diimplementasikan ke program dalam bahasa Pascal versi 1.5. Data instance yang diberikan untuk kedua algoritma tersebut adalah profit p_j berdistribusi secara random dalam [1,70], bobot w_j berdistribusi dalam [1,40] dengan kapasitas $b(.)$. Fungsi $b(.)$ dikonstruksikan dengan membangun n bilangan random dalam [1,40] dan memilih nilai ini dalam urutan tidak bertambah. Untuk setiap nilai n diberikan tiga data instance dan setiap data instance dijalankan pada kedua program. Kemudian waktu tempuhnya dicatat dan dihitung waktu tempuh rata-ratanya.

2. MASALAH KNAPSACK COLLAPSING

MKC 0-1 didefinisikan sebagai berikut :

Memaksimumkan $\sum_{i=1}^{n} p_i x_i$

dengan kendala $\sum_{i=1}^{n} w_i x_i \le b\left(\sum_{i=1}^{n} x_i\right), \quad x_i \in \{0,1\}, i = 1,...,n$

p_i, w_i, x_i bilangan bulat positif

p_i : "profit" item *i*

w_i : "bobot" item *i*

x_i : variabel keputusan

b(i) : fungsi tidak naik atas{1,2,..,n} yang menyatakan kapasitas knapsack.

MKC akan diselesaikan dengan reduksi ke masalah knapsack baku (RMKB) dan pendekatan program dinamik (PPD).

Reduksi ke Masalah Knapsack Baku

Diambil $A = \sum_{i=1}^{n} w_i$ dan $C = \sum_{i=1}^{n} p_i$. Tanpa menghilangkan keumuman diasumsikan $0 \le b(i) \le A$ untuk semua *i*, dengan kata lain diambil nilai *b(i)* adalah *A* dan $n \ge 3$. Akan dibangun MKB dengan 2*n* item yang mempunyai bobot $\alpha_i,...,\alpha_{2n}$ dan nilai $\gamma_1,...,\gamma_{2n}$ untuk item yang bersesuaian. Didefinisikan bobot baru

$$\alpha_i = w_i + A, i = 1,..,n$$
$$\alpha_i = (4n - i)A - b(i - n), i = n + 1,..,2n$$

dan nilai baru

$$\gamma_i = p_i + C, i = 1,..,n$$
$$\gamma_i = (3n + 1 - i)C, i = n + 1,..,2n$$

Item dengan indeks {1,..,*n*} disebut item kecil dan item dengan indeks {*n*+1,..,2*n*} disebut item besar. Pada MKB ini maksimum jumlah item yang dimuat adalah *n* + 1 item yaitu *n* item kecil dan satu item besar. Jadi kapasitas knapsack pada MKB adalah

$$c = \sum_{i=1}^{n}(w_i + A) + (4n - 2n)A - b(2n - n)$$
$$= A + nA + 2nA - A = 3nA$$

MKB adalah menentukan himpunan bagian item yang memaksimumkan nilai dengan bobot terbesar *c*. Diperoleh MKB Memaksimumkan $\sum_{i=1}^{2n} \gamma_i x_i$

dengan kendala $\sum_{i=1}^{2n} \alpha_i x_i \le c$, $x_i \in \{0,1\}, i = 1,..,2n$

MKB ini diselesaikan dengan algoritma minimal untuk masalah knapsack 0-1 (algoritma minknap). Algoritma minknap didasarkan pada program dinamik dengan batas waktu $O(nc)$. Karena $c=3nA$ maka MKC diselesaikan melalui reduksi ke MKB dengan batas waktu $O(n^2A)$. Dari reduksi ini diperoleh lemma dan teorema di bawah ini.

Lemma 1

Dalam penyelesaian fisibel MKB, knapsack memuat paling banyak satu item besar. Dalam kasus indeks item besar j, $n+1 \le j \le 2n$ *knapsack memuat paling banyak* $j-n$ *item kecil.*

Bukti

Item besar j, $n+1 \le j \le 2n$ mempunyai bobot $\alpha_j = (4n-j)A - b(j-n)$.

Untuk $j=2n$ diperoleh

$$\alpha_j = (4n-2n)A - b(2n-n) = (2n-1)A$$

dan untuk $j=n+1$ diperoleh

$$\alpha_j = (4n-n-1)A - b(n+1-n) = (3n-1)A - b(1)$$

Karena $b(1) \le A$ dan $b(1) \ge 0$ maka berlaku

$$(2n-1)A \le \alpha_j \le (3n-1)A, n+1 \le j \le 2n.$$

Karena $c=3nA$ maka knapsack tidak mungkin memuat dua atau lebih item besar. Untuk membuktikan pernyataan kedua dalam lemma, diambil item besar dengan indeks j termuat dalam knapsack . Kapasitas yang tersisa untuk item kecil adalah

$$3nA - \alpha_j = 3nA - (4n-j)A + b(j-n) = (j-n)A + b(j-n)$$

Karena setiap item mempunyai bobot paling sedikit A dan $b(j-n) \le A$ maka paling banyak $j-n$ item kecil dapat termuat.

Lemma 2

Dalam penyelesaian fisibel MKB dengan nilai sasaran terkecil $(2n+1)C+1$, *knapsack memuat sebuah item besar dengan indeks j,* $n+1 \le j \le 2n$ *dan tepatnya j-n* item kecil.

Bukti

Nilai keseluruhan untuk semua item kecil adalah $\sum_{i=1}^{n}\gamma_i = \sum_{i=1}^{n}(w_i + C)$. Untuk mencapai nilai $(2n+1)C+1$ maka item besar dengan indeks j harus termuat dalam knapsack. Diambil I himpunan indeks item kecil yang termuat dalam knapsack, $|I|$=banyaknya himpunan I. Andaikan $|I| < j-n$, maka nilai keseluruhan dalam knapsack paling besar

$$\gamma_j + \sum_{i\in I}(C+w_i) = (3n+1-j)C + |I|C + \sum_{i\in I} w_i$$
$$\leq (3n+1-j)C + (j-n-1)C + C = (2n+1)C$$

Karena diketahui nilai sasaran terkecil $(2n+1)C+1$ maka terjadi kontradiksi. Jadi paling sedikit $j-n$ item kecil harus termuat dalam knapsack. Dengan lemma 1 terbukti tepat $j-n$ item kecil termuat dalam knapsack.

Teorema 1

MKC mempunyai penyelesaian fisibel dengan nilai sasaran V jika dan hanya jika MKB mempunyai penyelesaian fisibel dengan nilai sasaran $V+(2n+1)C$.

Bukti

Diketahui MKC mempunyai penyelesaian fisibel dengan nilai sasaran V. Didefinisikan $I = \{i \mid x_i = 1\}$. Dengan kata lain $\sum_{i\in I} p_i = V$ dan $\sum_{i\in I} w_i \leq b(|I|)$.

Diberikan penyelesaian untuk MKB memuat semua item kecil dengan indeks dalam I dan item besar dengan indeks $n+|I|$. Nilai sasaran pada penyelesaian ini adalah

$$\gamma_{n+|I|} + \sum_{i\in I}(C+p_i) = (3n+1-n-|I|)C + |I|C + \sum_{i\in I} p_i$$
$$= (2n+1-|I|)C + |I|C + V$$
$$= (2n+1)C + V$$

dan bobotnya adalah

$$\alpha_{n+|I|} + \sum_{i\in I}\alpha_j = (4n-n-|I|)A - b(n+|I|-n) + \sum_{i\in I}(w_i + A)$$

$$= 3nA - |I|A - b(|I|) + \sum_{i \in I} w_i + |I|A$$
$$= 3nA + \sum_{i \in I} w_i - b(|I|)$$

Karena $\sum_{i \in I} w_i \le b(|I|)$ maka diperoleh $3nA + \sum_{i \in I} w_i - b(|I|) \le 3nA = c$. Terbukti MKB mempunyai penyelesaian fisibel dengan nilai sasaran $V + (2n+1)C$.

Diketahui MKB mempunyai penyelesaian fisibel dengan nilai sasaran $V + (2n+1)C$. Lemma 2 menyatakan bahwa knapsack memuat sebuah item besar dengan indeks j, $n+1 \le j \le 2n$ dan $j-n$ item kecil . Diambil I himpunan indeks item kecil dalam knapsack, $|I| = j-n$. Keseluruhan bobot pada penyelesaian fisibel untuk MKB paling besar adalah *c*, artinya

$$c = 3nA \ge \alpha_j + \sum_{i \in I} \alpha_i = (4n - j)A - b(j-n) + \sum_{i \in I} (A + w_i)$$
$$= (4n - n - |I|)A - b(n + |I| - n) + \sum_{i \in I} A + \sum_{i \in I} w_i$$
$$= 3nA - |I|A - b(|I|) + |I|A + \sum_{i \in I} w_i$$
$$= 3nA + (\sum_{i \in I} w_i - b(|I|))$$

Ketidaksamaan ini bernilai benar. Nilai sasaran pada penyelesaian ini adalah

$$(2n+1)C + V = \gamma_j + \sum_{i \in I} \gamma_i$$
$$= (3n + 1 - j) + \sum_{i \in I} (C + p_i)$$
$$= (3n + 1 - |I| - n)C + \sum_{i \in I} C + \sum_{i \in I} p_i$$
$$= (2n+1)C - |I|C + |I|C + \sum_{i \in I} p_i$$
$$= (2n+1)C + \sum_{i \in I} p_i$$

Dari kesamaan ini diperoleh $V = \sum_{i \in I} p_i$. Jadi item pada MKC yang mempunyai indeks dalam I mempunyai nilai sasaran V dan bobot paling besar $b(|I|)$.

JURNAL MATEMATIKA DAN KOMPUTER
Vol. 5. No. 3, 168 - 177, Desember 2002, ISSN : 1410-8518

Pendekatan Program Dinamik

Selain dengan cara reduksi di atas MKC dapat diselesaikan dengan pendekatan program dinamik (PPD). Diberikan b' adalah nilai maksimum dari $b(i), i = 1,...,n$. Didefinisikan $f_{i,k}(\tilde{c}); i = 1,..,n; k = 1,..,i; \tilde{c} = 0,..,b(k)$ menjadi penyelesaian optimal untuk masalah knapsack dengan dua kendala yang didefinisikan pada i item pertama.

$$f_{i,k}(\tilde{c}) = maks\left\{\sum_{j=1}^{i} p_j x_j \mid \sum_{j=1}^{i} w_j x_j \le b(k); \sum_{j=1}^{i} x_j = k; x_j \in \{0,1\},\ j = 1,..,i\right\}$$

Diambil himpunan nilai awal sebagai berikut :

$f_{0,0}(\tilde{c}) = 0$ untuk semua $\tilde{c} = 0,..,b'$

$f_{0,k}(\tilde{c}) = -\infty$ untuk semua $k = 1,...,n$ dan semua $\tilde{c} = 0,..,b'$

Mudah dipahami bahwa $f_{i,k}(\tilde{c})$ mempunyai interpretasi nilai optimal pada rekursi ke-i dengan i item yang tersedia tetapi jumlah item yang dimuat sebanyak k dan $\tilde{c}$ menyatakan batas bobot yang mungkin , jadi $\tilde{c}$ tidak selalu tunggal. Jelas bahwa $f_{i,k}(\tilde{c})$ tidak fisibel jika $k > i$, karena tidak mungkin knapsack memuat sebanyak item yang lebih besar dari item yang tersedia. Nilai $f_{i,k}(\tilde{c})$ ditentukan dengan relasi rekursi

$$f_{i,k}(\tilde{c}) = maks\begin{cases} f_{i-1,k}(\tilde{c}) \\ f_{i-1,k-1}(\tilde{c} - w_i) + p_i, k > 0, \tilde{c} - w_i > 0 \end{cases}$$

Nilai optimal akan diperoleh pada rekursi ke-n yaitu $Z = \underset{k=1,...,n}{maks} f_{n,k}(b(n))$

Pada setiap tahapan i diperlukan sebanyak k perbandingan dan penghitungan dan untuk suatu k diperlukan sebanyak $b(k)$ perbandingan dan penghitungan . Karena nilai maksimum k adalah n, nilai maksimum $b(k)$ adalah b' dan i berjalan dari 1 sampai n maka algoritma PPD mempunyai batas waktu $O(n^2 b')$.

Pada kenyataannya proses rekursi tidak menggunakan monotonisitas $b(.)$, jadi pendekatan program dinamik dan reduksi ke MKB berlaku untuk sebarang distribusi fungsi kapasitas.

Dengan menggunakan PPD diperlukan storage untuk menyimpan state (π,μ) dengan $\pi = f_{i,k}(\mu)$. Jika $b(.)$ fungsi kapasitas dengan urutan tidak naik maka diperoleh proposisi di bawah ini.

Proposisi 1

Batas atas pada state (π,μ) *dengan* $\pi = f_{i,k}(\mu)$ *adalah* $u(\pi,\mu) = \lfloor \pi + (b(k+1) - \mu)\, p_{i+1} / w_{i+1} \rfloor$

Bukti

Sesuai dengan definisi state (π,μ) yang bersesuaian dengan vektor penyelesaian dengan $\sum_{j=1}^{i} x_j = k$, maka untuk memperbaiki penyelesaian knapsack harus memuat $k+1$ atau lebih item. Batas muncul sebagai penyelesaian untuk masalah maks$\left\{\pi + \sum_{i=k+1}^{n} p_i \mid \mu + \sum_{i=k+1}^{n} w_i \le b(k+1); x_i \ge 0\right\}$.

Proposisi 2

Diambil $h = maks\left\{j \mid \sum_{i=1}^{j} w_i \le b(j)\right\}$ *dan* $K = maks\left\{j \mid b(j) \ge \sum_{i=1}^{h} w_i\right\}$, *maka* $\sum_{i=1}^{n} x_i \le K$ *dalam sebarang penyelesaian optimal untuk MKC.*

Bukti

Diambil $B = \sum_{i=1}^{h} w_i$. Penyelesaian optimal untuk masalah knapsack 0-1: $maks\left\{Z = \sum_{i=1}^{n} p_i \mid \sum_{i=1}^{n} w_i \le c; x_i \in (0,1)\right\}$ diberikan oleh $Z = \sum_{i=1}^{h} p_i$. Nilai Z ini juga menjadi penyelesaian fisibel untuk MKC. Untuk masalah knapsack 0-1 yang didefinisikan pada item yang sama tetapi dengan kapasitas yang lebih kecil $b(k) < c, k = K+1, \cdots, n$ dan dengan menambah kendala $\sum_{i=1}^{n} x_i = k$ maka nilai sasaran tidak lebih baik dari *Z*. Dengan kata lain $\sum_{i=1}^{n} x_i \le K$.

Vektor penyelesaian MKC yang diselesaikan dengan PPD diperoleh dengan menelusuri kembali himpunan $f_{i,k}(\tilde{c})$ yang menghasilkan penyelesaian optimal *Z*. Proses penelusuran dikerjakan dengan cara sebagai berikut :

- Menentukan nilai *k* yaitu nilai yang memberikan $f_{n,k}(b(k)) = Z$

- Menentukan nilai biner x_i

  for l:= n downto 1 do

  if $f_{l,k}(b(k)) = f_{l-1,k-1}(b(k) - w_l) + p_l$ then

  $b(k) := b(k) - w_l$

  k := k-1;

  x_l :=1

  else

  k :=k ;

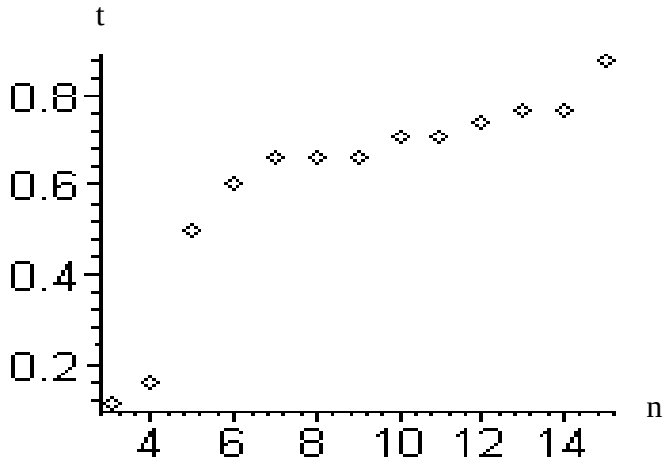
  x_l :=0

Eksperimen

Kedua program dijalankan pada komputer AT 486. Dari eksperimen diperoleh waktu tempuh rata-rata kedua algoritma dalam menyelesaikan suatu masalah berukuran n

	Waktu tempuh (detik)	
n	RMKB	PPD
3	0.11	0
4	0.16	0
5	0.50	0
6	0.60	0
7	0.66	0
8	066	0
9	0.66	0
10	0.71	0
11	0.71	0
12	0.74	0
13	0.77	0
14	0.77	0
15	0.88	0

Tabel waktu tempuh rata-rata



Grafik fungsi waktu tempuh algoritma RMKB

3. KESIMPULAN

1. Masalah knapsack collapsing dapat diselesaikan dengan reduksi ke masalah knapsack baku (RMKB) dalam $O(n^2A)$ dan pendekatan program dinamik (PPD) dalam $O(n^2b')$.
2. Algoritma RMKB dan PPD termasuk algoritma yang efisien karena fungsi kompleksitas waktunya adalah pseudo-polinomial.
3. Waktu tempuh algoritma PPD lebih kecil daripada waktu tempuh algoritma RMKB dalam menyelesaikan masalah karena nilai b' selalu lebih kecil sama dengan A.

DAFTAR PUSTAKA

1. Pferschy U, Pisinger D and Woeginger G. J, Simple but Efficient Approaches for the Collapsing Knapsack Problem, Bericht Nr. 32, *Optimierung und Kontrolle*, Institute of Mathematics T. U. Graz, Austria, 1995.
2. Pisinger, D, *Algorithms for Knapsack Problems*, Ph. D. Thesis Report, Dept. of Computer Science, University of Copenhagen, Denmark, 1995.
3. Posner, M. E and Guignard, M, *The Collapsing 0-1 Knapsack Problem*, *Mathematical Programming*, 1978, 15 : 155-161.